

Políticas de nomenclatura para el equipo de desarrollo de ACME

Objetivo:

Estandarizar la nomenclatura utilizada por el equipo de desarrollo de ACME para bases de datos, variables, funciones, clases y proyectos Git, con el fin de mejorar la legibilidad, mantenibilidad y coherencia del código.

Alcance:

Estas políticas aplican a todo el código desarrollado por el equipo de ACME, incluyendo proyectos nuevos y existentes.

Políticas:

Bases de datos:

- Los nombres de las bases de datos deben ser descriptivos y reflejar su propósito.
- Se recomienda utilizar nombres en minúscula y separados por guiones bajos.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.
- Ejemplos: `base_de_datos_clientes`, `inventario_productos`.

Tablas:

- Los nombres de las tablas deben ser descriptivos y reflejar la entidad que representan.
- Se recomienda utilizar nombres en plural y minúscula, separados por guiones bajos.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.
- Ejemplos: `clientes`, `productos`, `pedidos`.

Columnas:

- Los nombres de las columnas deben ser descriptivos y reflejar el dato que almacenan.
- Se recomienda utilizar nombres en minúscula y separados por guiones bajos.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.
- Se recomienda utilizar prefijos para indicar el tipo de dato (por ejemplo, `id`, `nombre`, `precio`, `fecha`).
- Ejemplos: `cliente_id`, `producto_nombre`, `pedido_fecha`.

Variables:

- Los nombres de las variables deben ser descriptivos y reflejar el valor que almacenan.
- Se recomienda utilizar camelCase.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.

- Ejemplos: `nombreCliente`, `precioProducto`, `esActivo`.

Funciones:

- Los nombres de las funciones deben ser descriptivos y reflejar la acción que realizan.
- Se recomienda utilizar camelCase.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.
- Se recomienda utilizar verbos en infinitivo.
- Ejemplos: `obtenerDatosCliente()`, `calcularTotalCompra()`, `enviarCorreoElectronico()`.

Clases:

- Los nombres de las clases deben ser descriptivos y reflejar la entidad que representan.
- Se recomienda utilizar PascalCase.
- Se deben evitar abreviaturas y acrónimos que no sean ampliamente conocidos.
- Ejemplos: `Cliente`, `Producto`, `Pedido`.

Ramas Git:

Siguiendo el flujo de trabajo Git Flow, se utilizan las siguientes ramas:

- `master`: Rama principal de producción. Debe estar siempre en un estado listo para ser publicado.
- `develop`: Rama de desarrollo principal. Contiene los últimos cambios estables y probados.
- `feature/*`: Ramas para implementar nuevas funcionalidades. Se crean a partir de `develop` y se fusionan en ella una vez terminadas y probadas.
- `hotfix/*`: Ramas para corregir errores urgentes. Se crean a partir de `master` y se fusionan en ella una vez corregido el error y probado.
- `release/*`: Ramas para preparar una nueva versión del software. Se crean a partir de `develop` y se fusionan en `master` una vez finalizadas las pruebas de la nueva versión.

Nomenclatura de las ramas:

- El nombre de la rama debe ser descriptivo y reflejar el propósito de la rama.
- Se recomienda utilizar guiones bajos para separar palabras.
- Se deben evitar espacios y caracteres especiales.
- Ejemplos: `feature/nueva-funcionalidad-login`, `hotfix/bug-conexion-base-datos`, `release/v1.0.0`.
-

Conclusión:

La adopción de estas políticas de nomenclatura ayudará a mejorar la calidad del código desarrollado por el equipo de ACME, lo que a su vez se traducirá en un mejor producto final para nuestros clientes.